

# Dealls! Backend Engineer Technical Test

Safiq Faray

[Syafiqfaray3@gmail.com](mailto:Syafiqfaray3@gmail.com) / +6281232158548

## Github URL

<https://github.com/Haruray/dealls-backend-test>

## Functional and Non-functional Requirements

### Functional Requirements

1. User Authentication
2. Swiping Mechanism
3. Swipe Limiting
4. User Verification Through Purchase

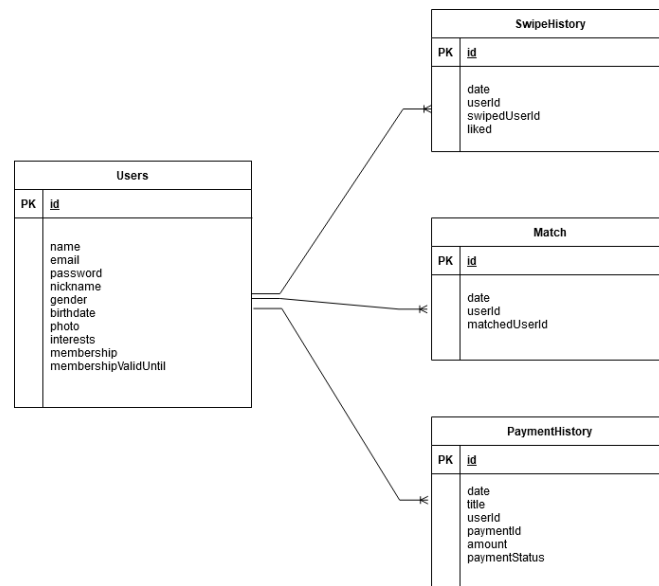
### Non-functional Requirements

1. Reliability
2. Maintainability
3. Test Coverage

## Tech Stack

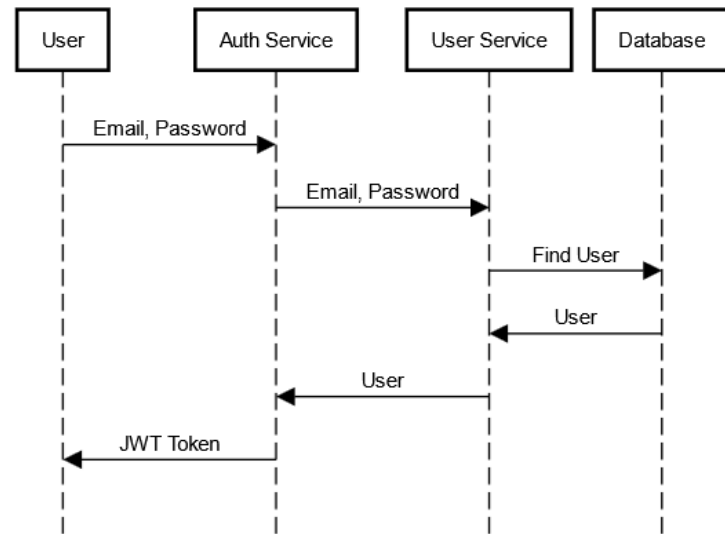
- Language : TypeScript
- Framework : NestJS  
NestJS is used because of its modular architecture, which makes it easy to build. Its built-in TypeScript support ensures type safety and reduces runtime errors, leading to more reliable code.
- Database : MySQL  
MySQL is used because its reliability and ease of use. It provides robust relational database features that are ideal for managing structured data, such as user profiles, swiping actions, and subscription details.

# ERD

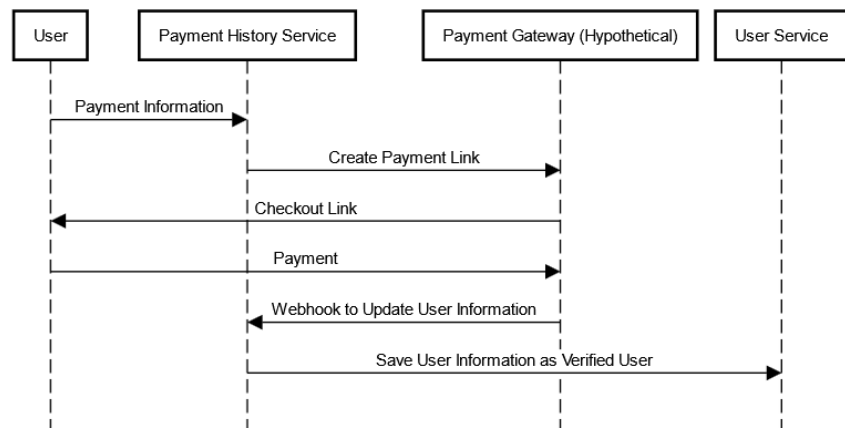


## Sequence Diagram

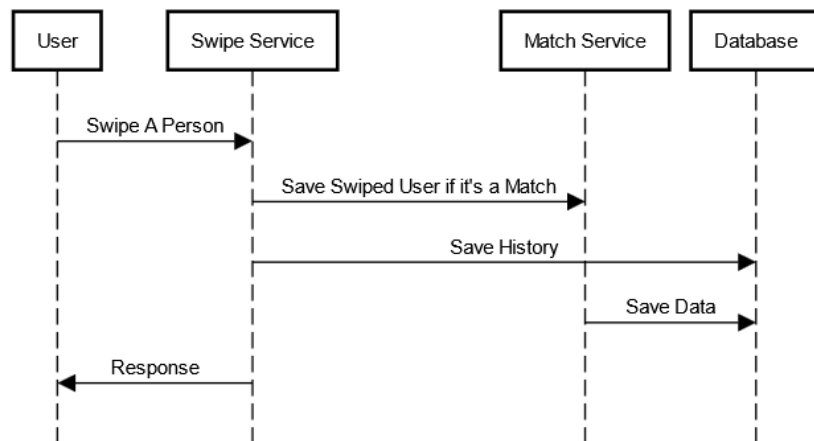
Authentication Sequence Diagram



Payment Diagram



Swipe Diagram



## Unit Tests

### MatchService Test

The MatchService test suite focuses on verifying the functionality of managing "matches" between users. The tests include:

- Definition Check: Ensures the MatchService is properly instantiated.
- Create Match: Validates the creation of a match between two users.
- Find All Matches: Checks if all existing matches can be retrieved.
- Find Match by ID: Tests retrieval of a specific match using its unique ID.
- Update Match: Ensures matches can be updated with new information.
- Remove Match: Verifies that a match can be deleted.

### PaymentHistoryService Test

The PaymentHistoryService test suite validates the functionality of handling payment history records. The tests include:

- Definition Check: Ensures the PaymentHistoryService is properly instantiated.
- Create Payment History: Confirms payment history creation for a specific user.
- Find All Payment Histories: Checks retrieval of all payment history records.
- Find Payment History by ID: Validates fetching a specific payment history using its ID.
- Update Payment History: Ensures payment histories can be updated with new details.
- Remove Payment History: Tests the deletion of a payment history record.

### SwipeHistoryService Test

The SwipeHistoryService test suite focuses on validating the functionality of swipe actions (e.g., "like" or "pass") between users. The tests include:

- Definition Check: Ensures the SwipeHistoryService is properly instantiated.
- Create Swipe History: Verifies the creation of a swipe history record between two users.
- Find All Swipe Histories: Tests the retrieval of all swipe history records.
- Find Swipe History by ID: Ensures a specific swipe history can be fetched using its ID.
- Update Swipe History: Validates the ability to update swipe history details (e.g., change a "like" to "pass").
- Remove Swipe History: Tests the deletion of a specific swipe history record.